



The
Patent
Office

INVESTOR IN PEOPLE

9899/3119

**PRIORITY
DOCUMENT**

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

REC'D 29 OCT 1999

WIPO PCT

09/787117

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

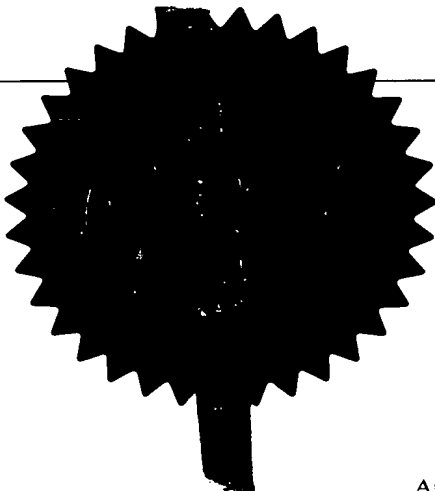
In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated

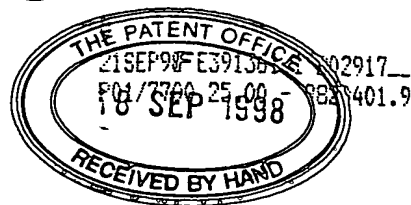
19 OCT 1999



This Page Blank (uspto)

Patents Form 1/77
Patents Act 1977
(Rule 16)

The
Patent
Office



Request for grant of a patent

The Patent Office
Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference
5271501/KB/RJD
2. Patent Application Number
18 SEP 1998 9820401.9
3. Full name, address and postcode of the or of each applicant (*underline all surnames*)
Software 2000 Ltd.
The Magdalen Centre
Oxford Science Park
Oxford
OX4 4GA

Patents ADP number (*if known*) 7515448001

If the applicant is a corporate body, give the country/state of its incorporation Country: England
State:

4. Title of the invention
Graphical User Interface

5. Name of agent Beresford & Co
"Address for Service" in the United Kingdom to which all correspondence should be sent 2/5 Warwick Court
High Holborn
London WC1R 5DJ

Patents ADP number 1826001

6. Priority details

Country Priority application number Date of filing

Patents Form 1/77

7. If this application is divided or otherwise derived from an earlier UK application give details

Number of earlier of application

Date of filing

8. Is a statement of inventorship and or right to grant of a patent required in support of this request? Yes

9. Enter the number of sheets for any of the following items you are filing with this form.

Continuation sheets of this form -



Description 36

Claim(s) 8

Abstract 1

Drawing(s) 14

10. If you are also filing any of the following, state how many against each item.

Priority documents -

Translations of priority documents -

Statement of inventorship and right to grant of a patent (*Patents form 7/77*) x1 + 4 copies

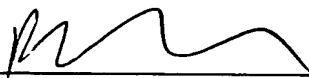
Request for preliminary examination and search (*Patents Form 9/77*) x 1

Request for Substantive Examination (*Patents Form 10/77*) -

Any other documents
(please specify)

11. I/We request the grant of a patent on the basis of this application

Signature



Date 18 September 1998

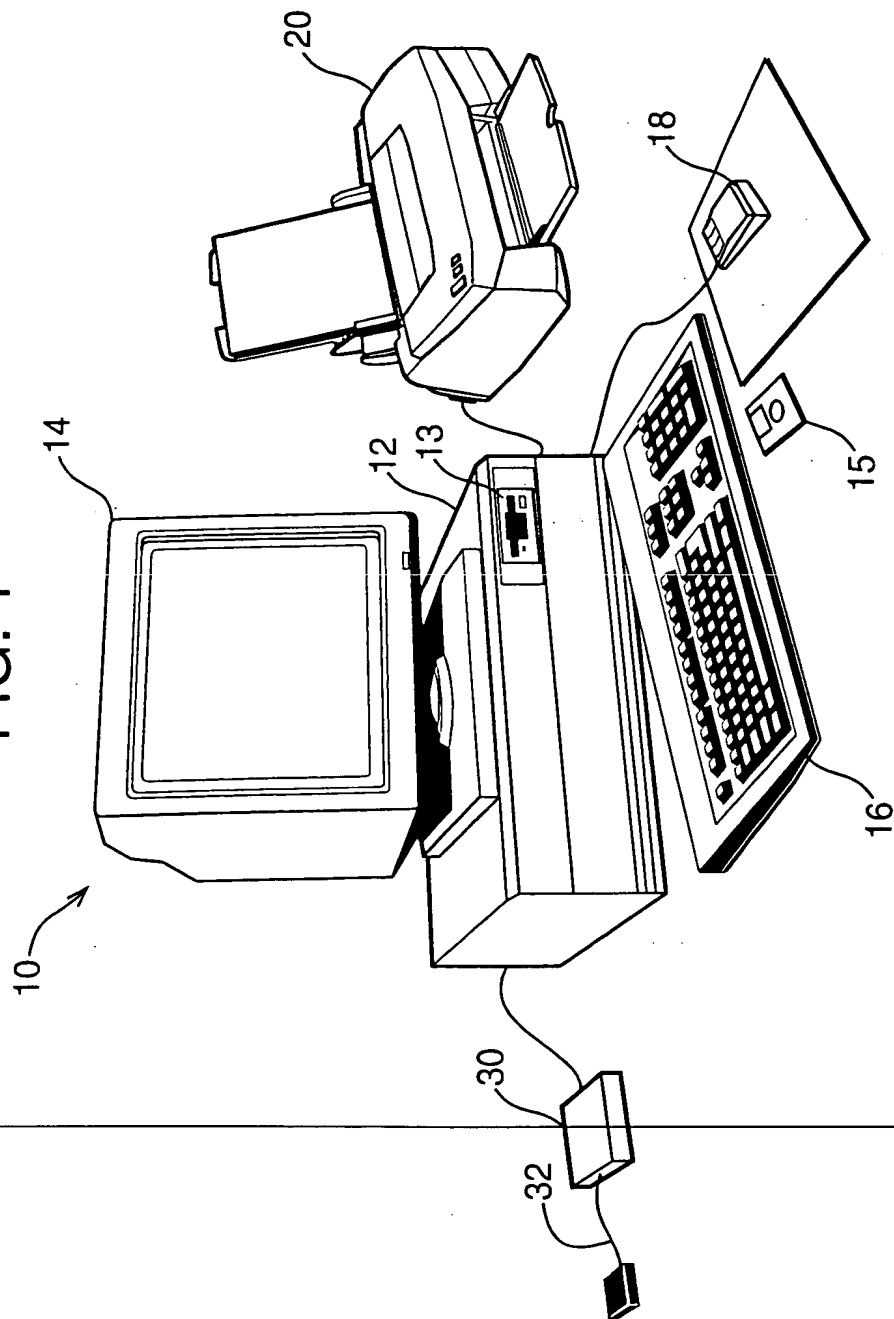
BERESFORD & Co

12. Name and daytime telephone number of person to contact in the United Kingdom

BERESFORD; Keith Denis Lewis

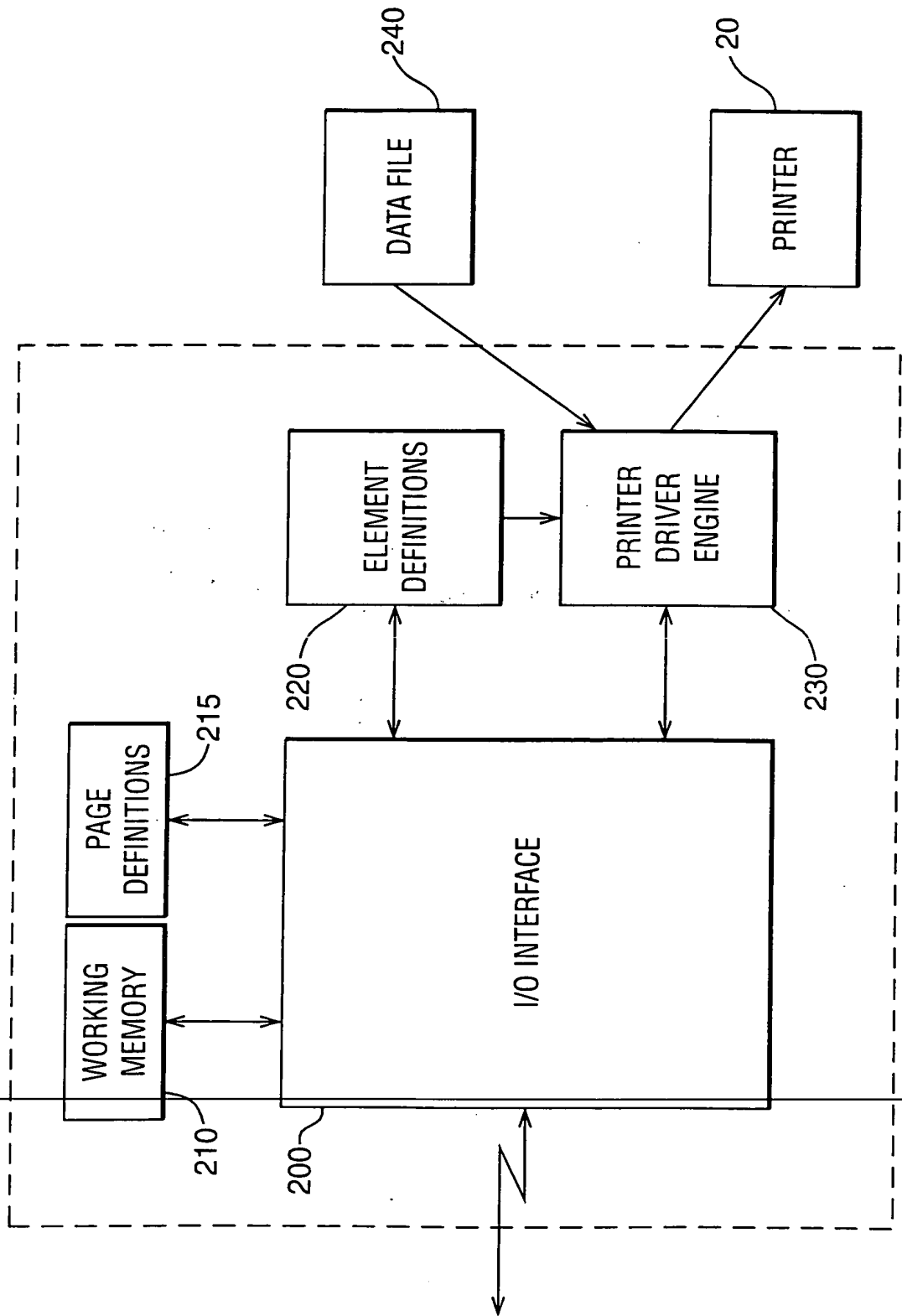
Tel:0171-831-2290

FIG. 1

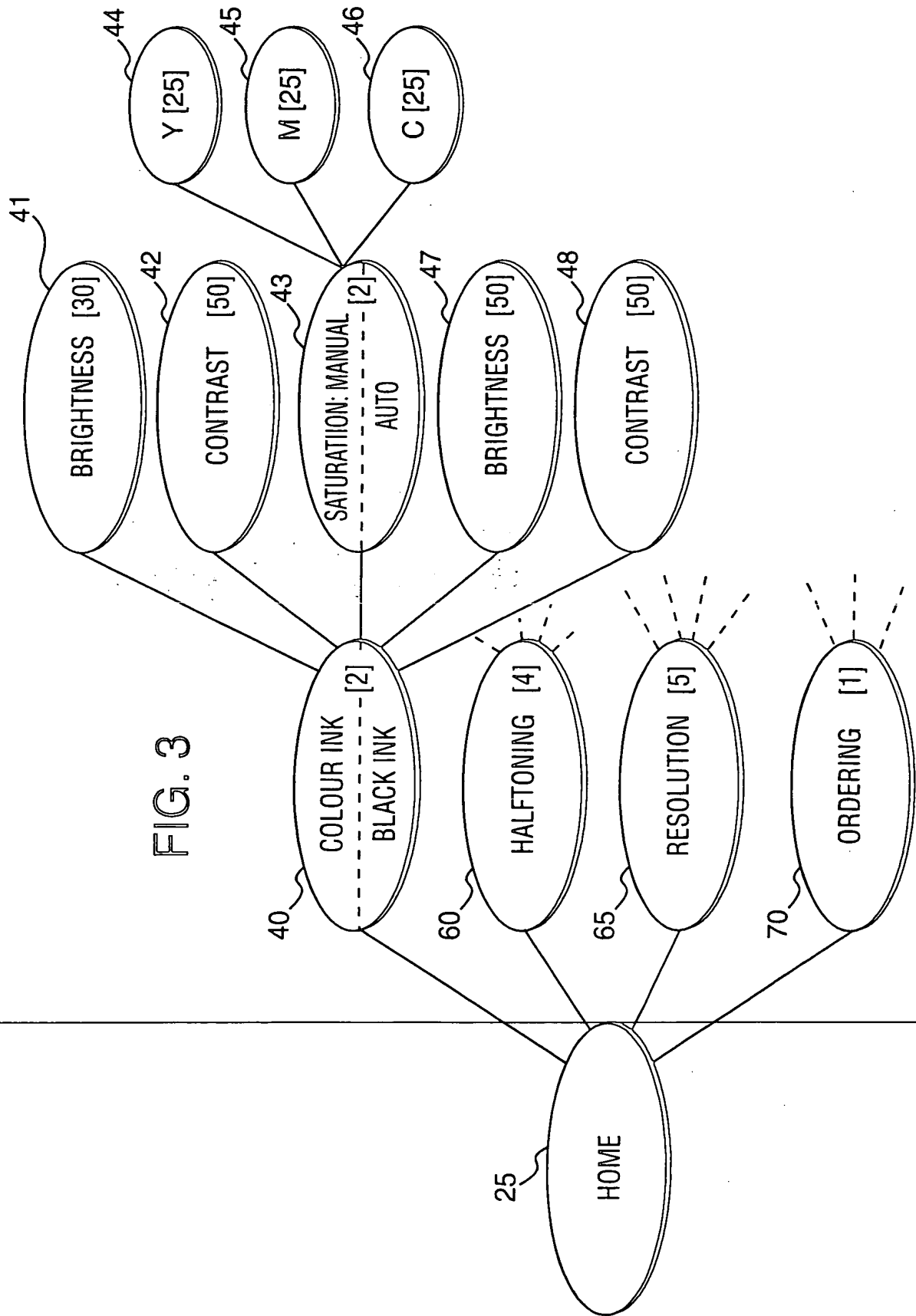


This Page Blank (uspto)

FIG. 2

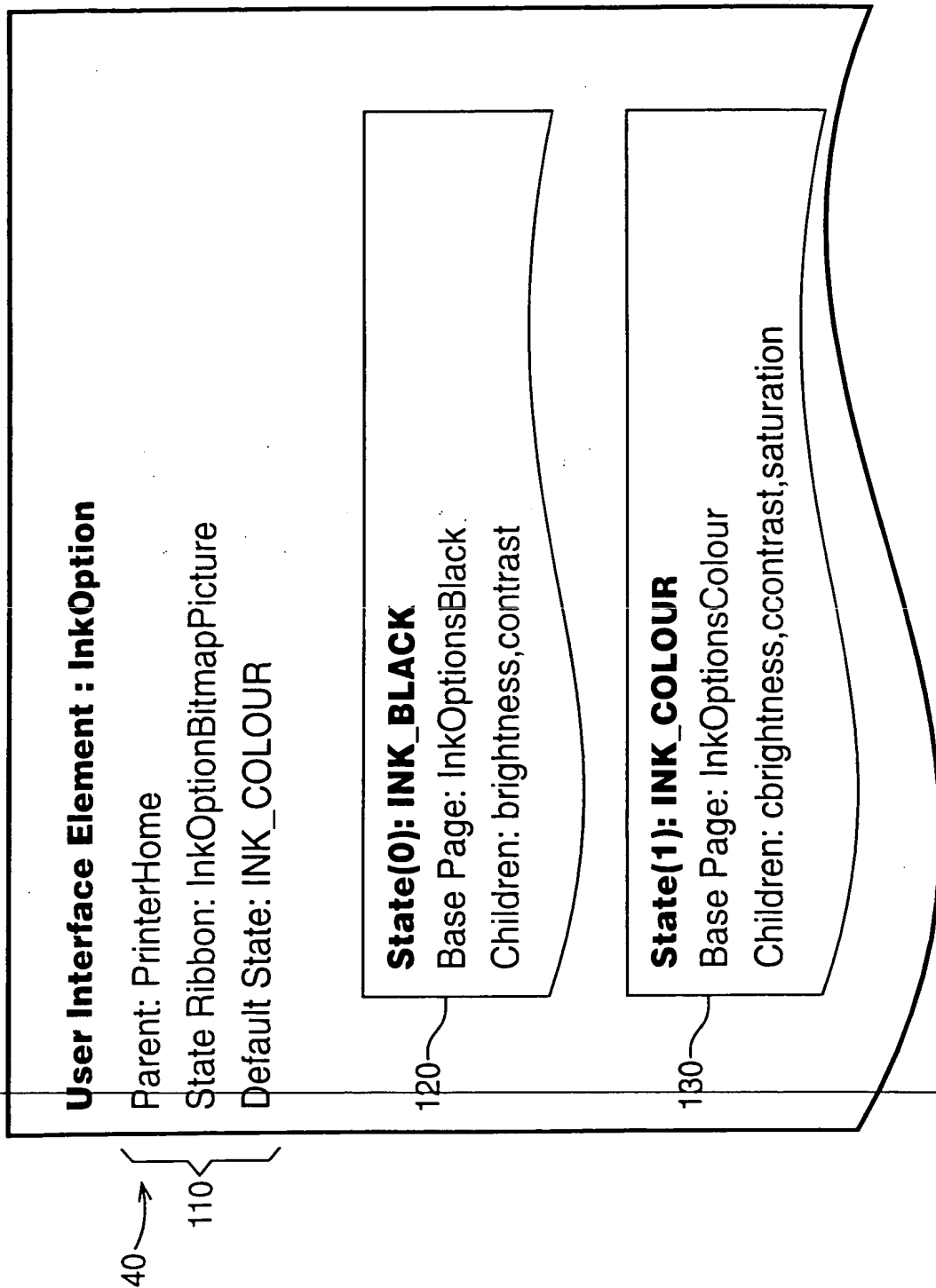


This Page Blank (uspto)



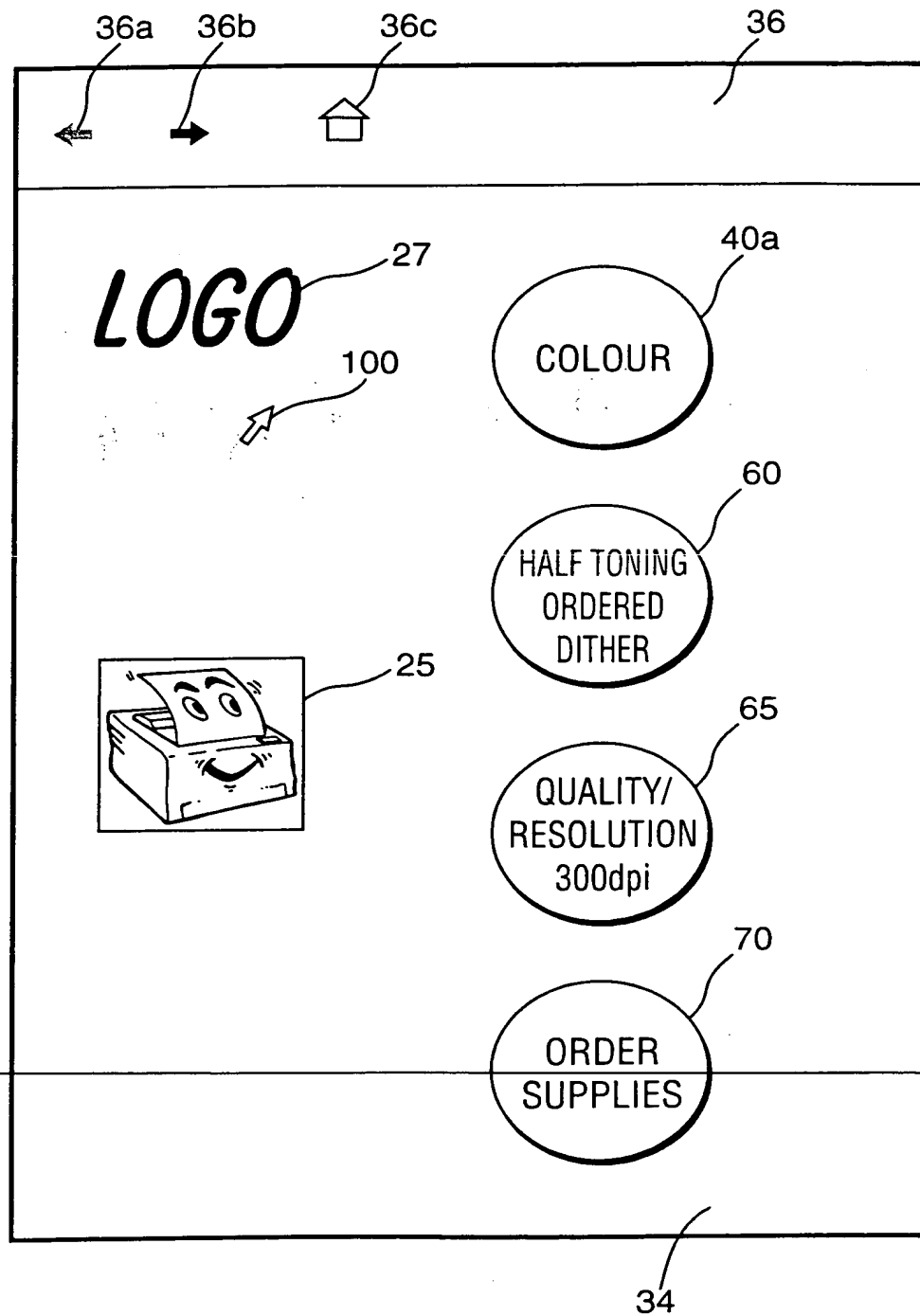
This Page Blank (uspto)

FIG. 4



This Page Blank (uspto)

FIG. 5



This Page Blank (uspto)

FIG. 6

Page : Homepage

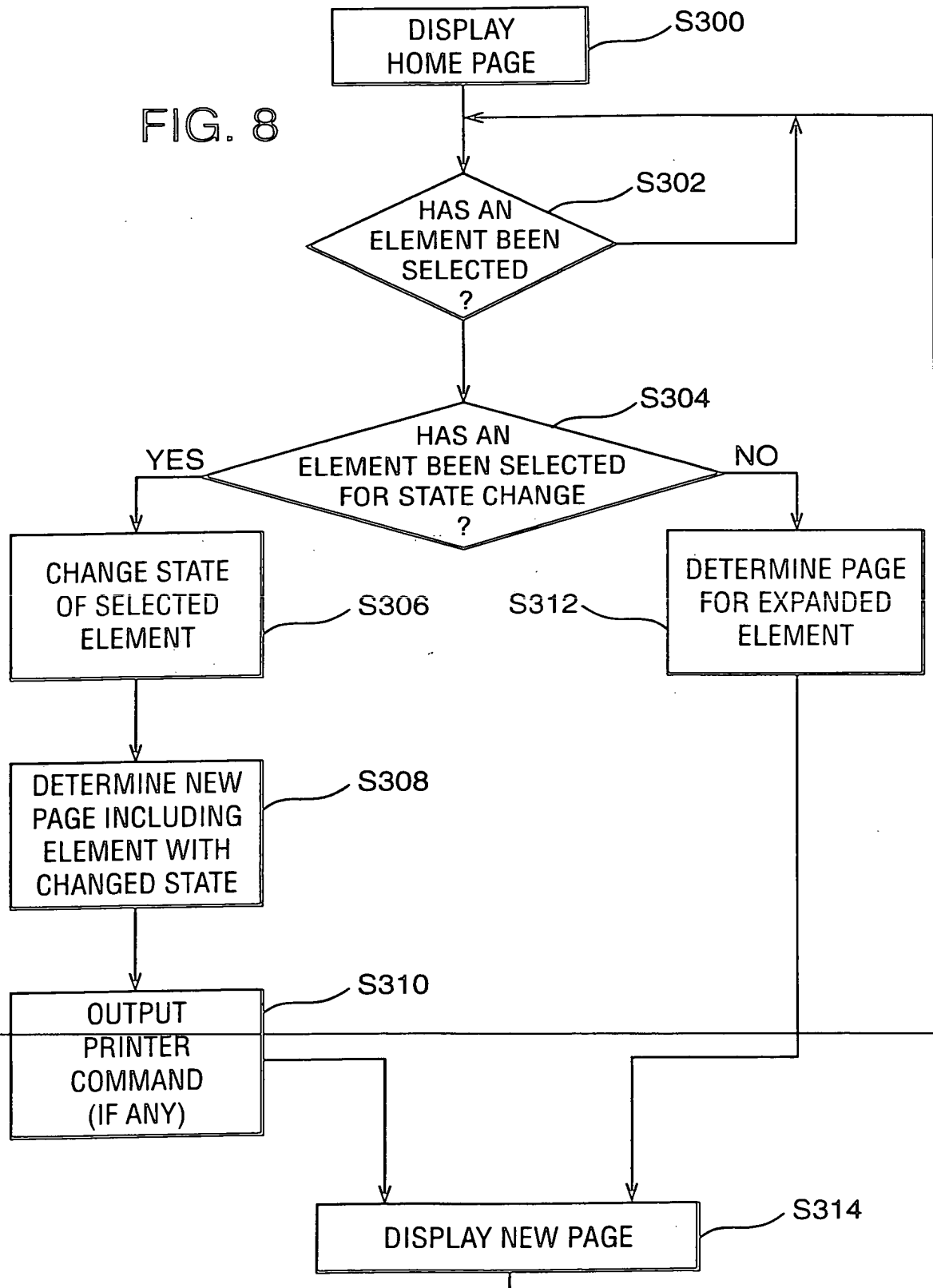
Element=CompanyLogo, 5,5, STATIC
Element=PrinterHome, 10.85, STATIC
Element=InkOption, 55,10, ACTIVE
Element=HalfToneOption, 55,60, ACTIVE
Element=ResOptions, 55,110, ACTIVE
Element=OrderSupplies, 55,160, ACTIVE

FIG. 7

PRESENT_PAGE	210
SELECTED_ELEMENT	211
	212
ELEMENT_0	213
ELEMENT_1	
ELEMENT_2	
ELEMENT_3	
⋮	

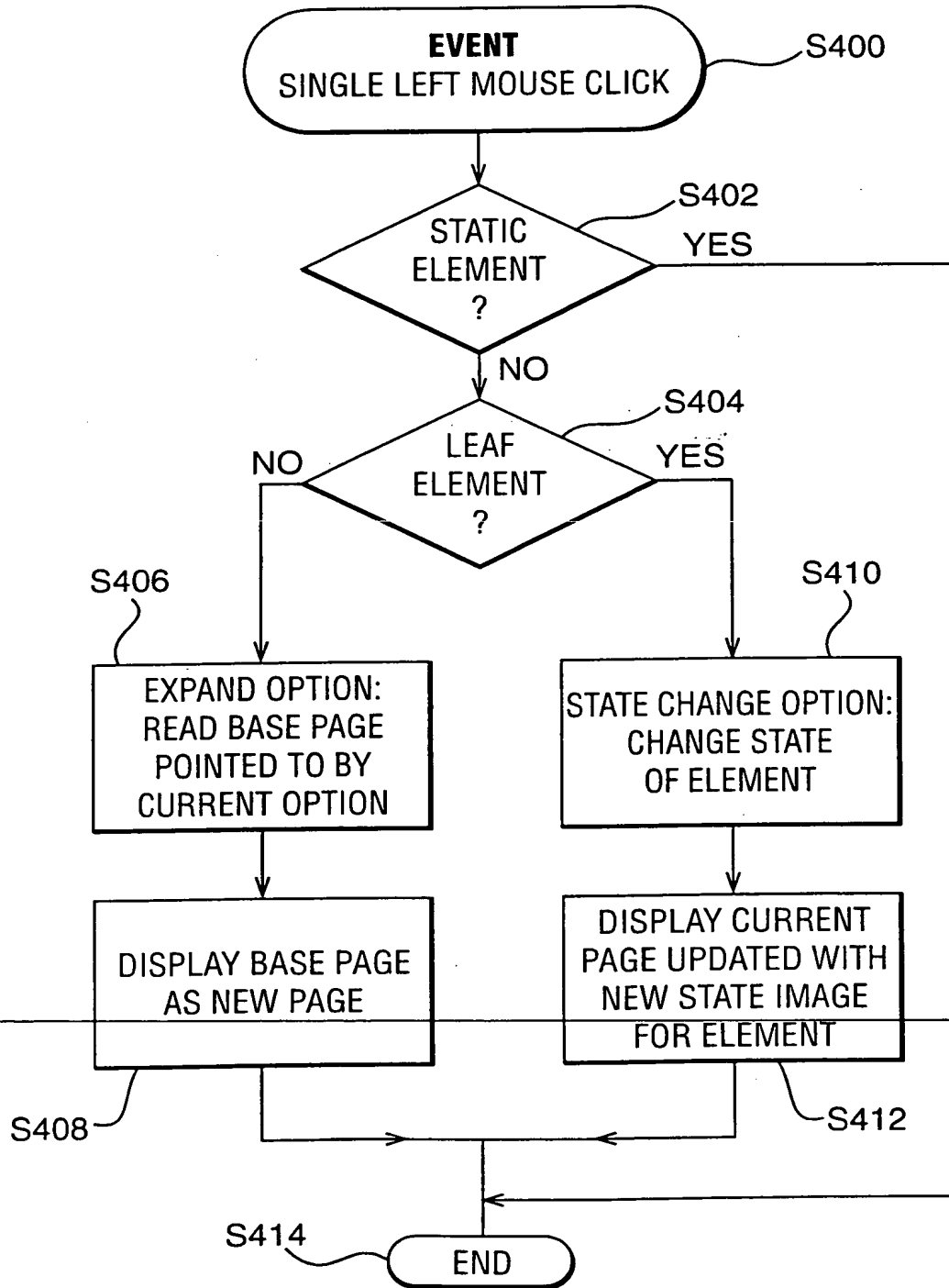
This Page Blank (uspto)

FIG. 8



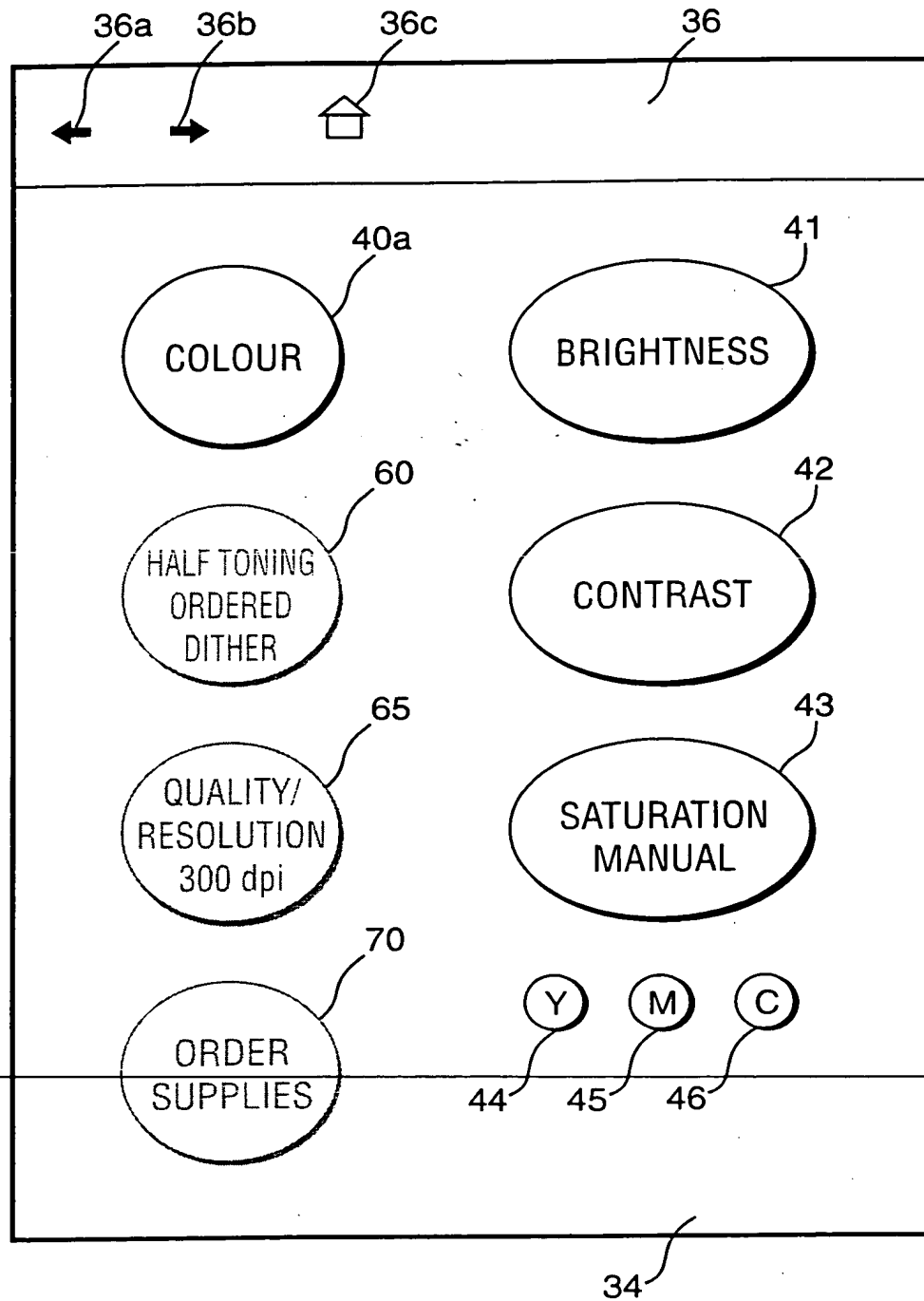
This Page Blank (uspto)

FIG. 9



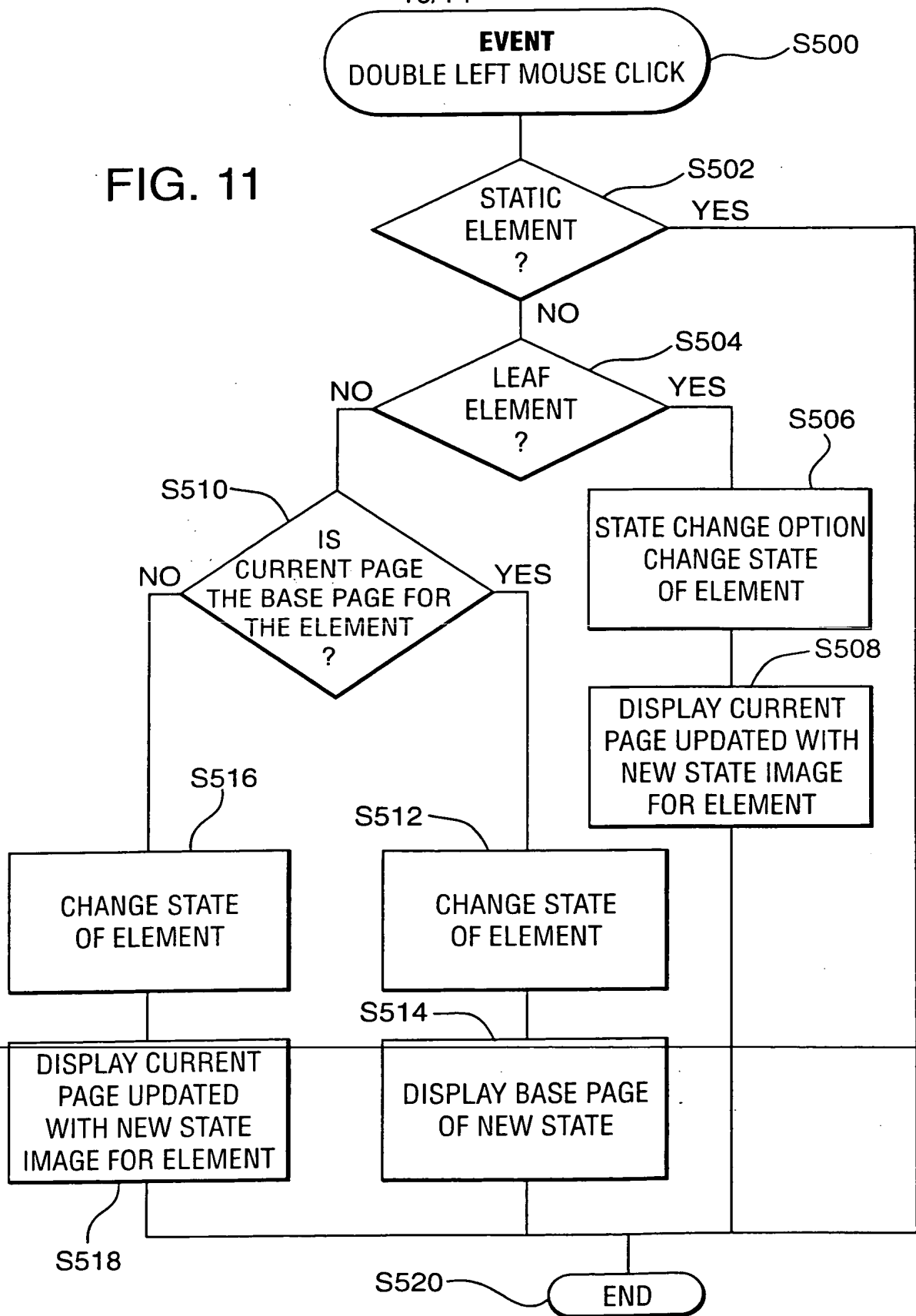
This Page Blank (uspto)

FIG. 10



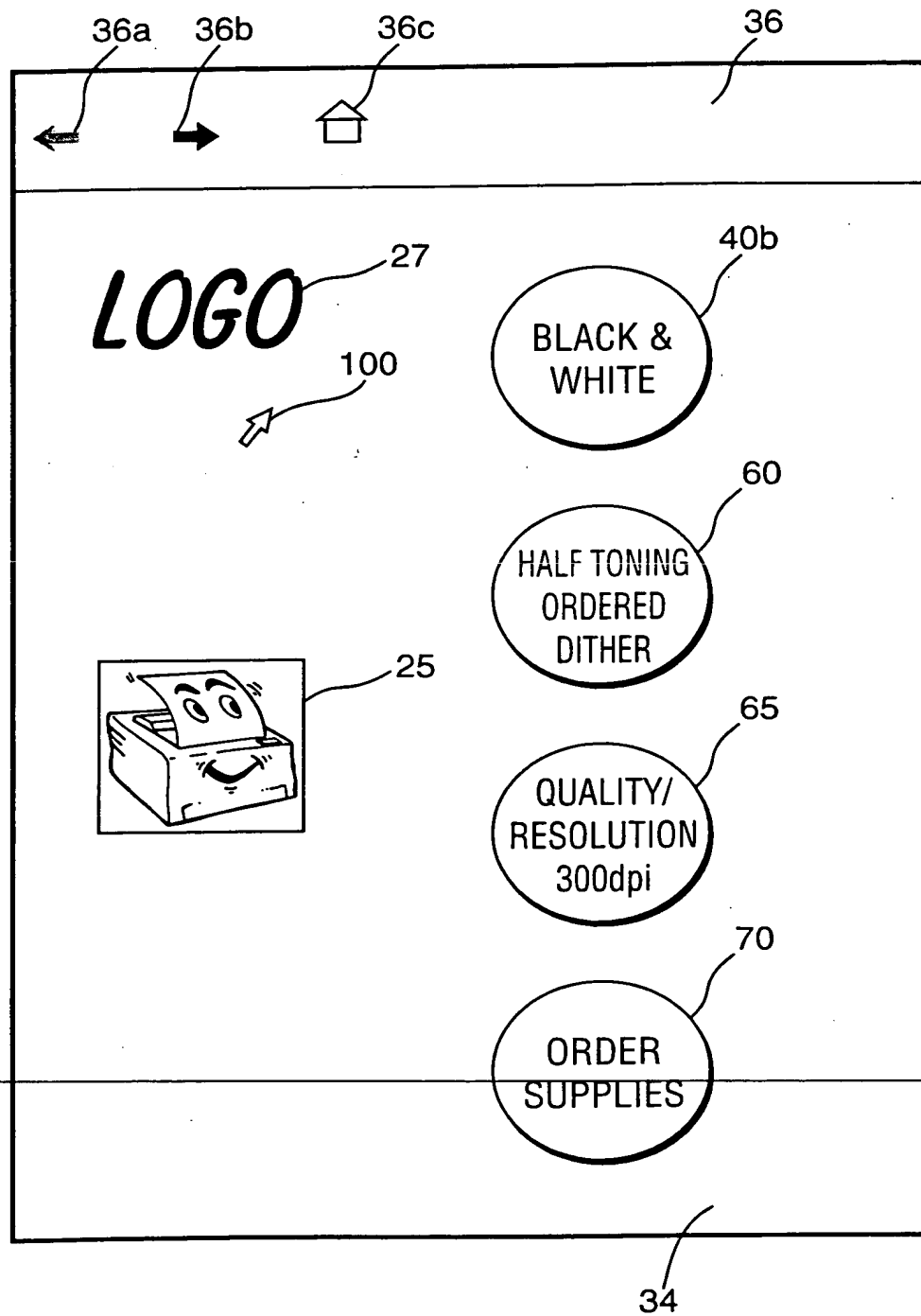
This Page Blank (uspto)

FIG. 11



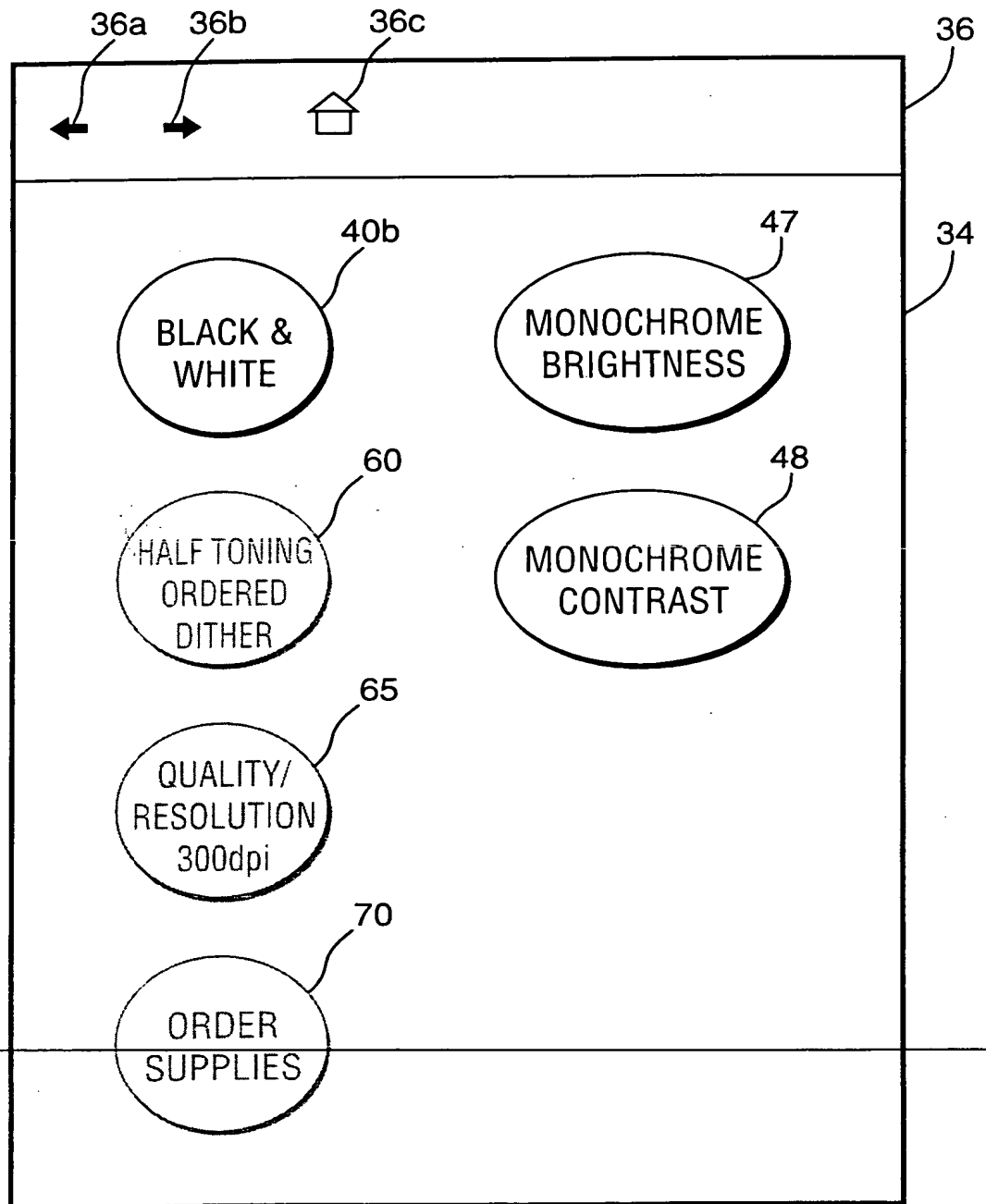
This Page Blank (uspto)

FIG. 12



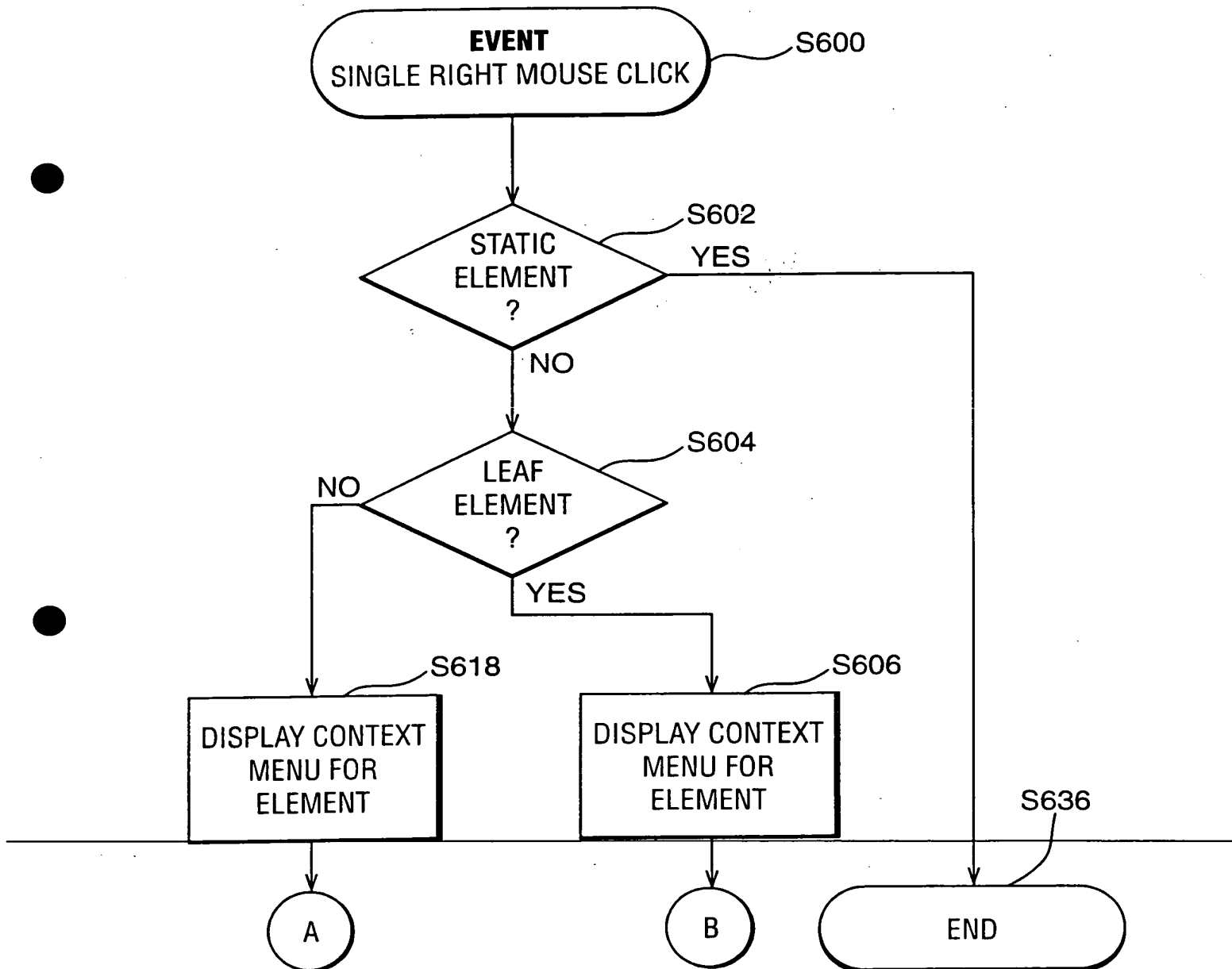
This Page Blank (uspto)

FIG. 13



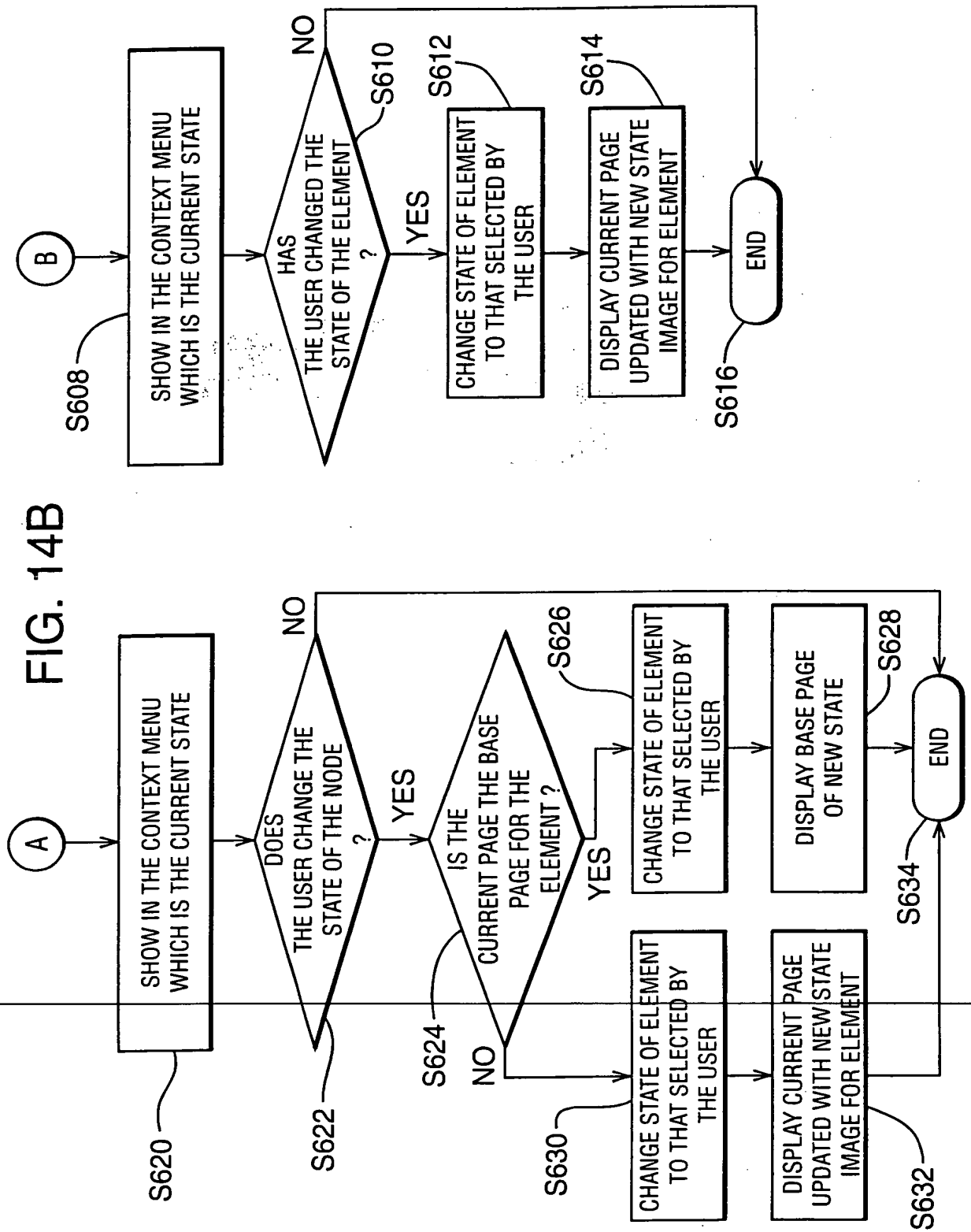
This Page Blank (uspto)

FIG. 14A



This Page Blank (uspto)

FIG. 14B



This Page Blank (uspic

GRAPHICAL USER INTERFACE

The present invention relates to device drivers for the interfacing of data between a personal computer and a peripheral and has particular, although not exclusive, application to printer drivers.

Personal computers may typically be interfaced to a number of peripherals such as printers, modems, scanners and the like. Each of these peripherals may be configurable in different manners (e.g. a printer may be able to print at various resolutions). Furthermore each peripheral may have different capabilities in comparison to peripherals of the same type (e.g. a printer may or may not be able to print in colour). It is therefore necessary, in certain cases, for the user to be able to specify the configuration of the peripheral for the particular job in hand. This may also require different processing of the data before it is output to a peripheral. The output of this user defined configuration information to the peripheral and the necessary processing of the data before outputting to the peripheral is carried out by the device driver.

As stated above, device drivers perform in essence two functions, the first being to convert data from the form used by the computer to a form usable by the driven

device. Their second function is to provide various command and control parameters to the driven device. Both the above functions are under the control of the user who communicates with the device driver via a user
5 interface.

The Windows (Trademark) Graphical User Interface (GUI) is now the most commonly used user interface for personal computers. This GUI may be by way of an adjunct to the
10 operating system (e.g. Windows 3.x and Windows 95/98) or be integral to the operating system (e.g. Windows NT). The Windows GUI causes controls to be displayed on the personal computer's display and these controls can be selected or varied using a pointing device, typically a
15 mouse, in conjunction with one or more buttons for confirming that the control pointed at is to be selected. These controls may include buttons, radio buttons or sliders.

20 However there is a lack of flexibility in a system designers choice of controls as a system designer is confined to a choice of a small number of different types

of controls in the design of the user interface for a device driver. This choice may restrict the
25 functionality that a designer is attempting to achieve. It also makes a device driver less "user friendly". Although a developer may create Windows custom controls,

these require a good working knowledge of Windows system programming.

A problem therefore arises as the present scheme of
5 constructing a device driver is complex, offers little
choice in how the display will appear to a user and the
architecture available means that such construction
requires very specialist skills.

10 The present invention addresses the above problem by
providing a device driver architecture which contains two
separate but related sets of data, the first set defining
the functionality of the features of the driver and the
second set, how those features will be presented by way
15 of graphical interface. Such an architecture provides
the designer of a device driver with increased
flexibility in the design of the appearance of the user
interface without affecting the functionality of the
driver.

20

Additionally with the advent of Windows 98 and the
Windows NT, it is desirable that the users view of the
system should be in the form of a browser interface such
that the distinction between the users view of the local
25 machine and the World Wide Web may become amalgamated
into one seamless view. The basis for defining such a
user interface would be a hypertext based system as used

over the internet. HTML (Hyper Text Mark-up Language) is such a system which defines certain user interface mechanisms and conventions and is presently used across many platforms including Windows 95/98, Windows NT and
5 UNIX based systems.

There remains a problem within the field of device drivers in that they continue to adhere to the legacy conventions of earlier operating systems and do not make
10 use of the facilities offered by the newer systems. HTML technology allows many user interfaces which each have their own look and feel to be created using the same underlying structure which is in itself heavily extensible. The invention brings the appearance of a
15 device driver interface into line with that which any user who has browsed the web will be familiar with.

The present invention provides a device driver for interfacing a personal computer to a peripheral, the
20 device driver comprising: a device driver engine for driving the peripheral in accordance with control
~~commands; means for defining a plurality of elements,~~
each element having at least one state, each state having an associated image for display to a user for user
25 selection and at least one of a pointer to one or more associated elements or a control command to be output to the device driver engine; means for defining a plurality

of pages, each page being associated with a state of an element and listing at least one of said one or more associated elements; and means for (i) receiving a user selection of a current state of a current element; (ii) 5 reading the page associated with the current state of the current element; (iii) generating a display using the associated image of each element identified in the page list for the current state of the current element; and (iv) outputting the control command, if any, of the 10 current state of the current element to the device driver engine.

An embodiment of the invention will now be described with reference to the accompanying figures in which:

15

Figure 1 shows a computer system including a personal computer and peripherals including a printer and a modem;

Figure 2 is a simplified block diagram of a printer 20 driver according to the embodiment of the present invention;

Figure 3 is a simplified diagram showing a hierarchical structure of a set of elements forming part of the 25 printer driver shown in Figure 2;

Figure 4 is a simplified diagram showing the data

structure defining the ink options element shown in Figure 3;

Figure 5 is a simplified diagrammatic representation of
5 a home page output as it would appear to the user of a
printer driver according to the embodiment of the
invention;

Figure 6 is a simplified diagrams showing the data
10 structure defining the home page;

Figure 7 is a simplified memory map of the working memory
of the I/O interface shown in Figure 2;

15 Figure 8 is a much simplified flow diagram which
illustrates the I/O interface of the printer driver
according to the embodiment of the invention;

Figure 9 is a flow diagram showing the events triggered
20 by a single click of the left mouse button;

Figure 10 is a simplified diagrammatic representation of
a screen output page as it would appear to the user
following a selection of the colour ink element of Figure
25 5 using a single click of the left mouse button;

Figure 11 is a flow diagram showing the events triggered

by a double click of the left mouse button;

Figure 12 is a simplified diagrammatic representation of a screen output page as it would appear to the user following a selection of the colour ink element of Figure 5 using a double click of the left mouse button;

Figure 13 is a simplified diagrammatic representation of a screen output page as it would appear to the user following a selection of the colour ink element of Figure 10 using a double click of the left mouse button; and

Figures 14A and 14B are a flow diagram of the events triggered by a single click of the right mouse button.

15

Figure 1 shows a personal computer 10 comprising a processor unit 12, a visual display unit (VDU) 14, a keyboard 16 and a pointing device which in this embodiment is a mouse 18. In this embodiment the mouse has two buttons to allow the user to confirm a selection, namely a right and a left button. The need for two buttons will be described in more detail below. The

processor unit 12 includes a floppy disc drive 13, which may read a floppy disc 13a in addition to the usual components of RAM, ROM, CPU, hard disc and the like (not shown). Connected to the personal computer system 10 are two peripherals, namely printer 20 and a modem 30. The

modem is connected to a telephone line (not shown) using cable 32. The personal computer system 10 can, therefore, communicate with outside devices such as the internet. In this embodiment, the printer 20 is a colour
5 ink jet printer.

The processor unit 12 is effective to run under the command of software instructions which are stored in the RAM of the unit. A group of these instructions form the
10 printer driver. This software may initially be provided upon a floppy disc 15 (or other storage medium) or downloaded from a data network (e.g. from the Internet) via modem 30.

15 Figure 2 is an overall block diagram of the printer driver 199 of this embodiment together with the printer 20 and a data file 240 to be printed. The printer driver engine 230 is a set of software instructions which instruct the CPU on the process which it must conduct in
20 order to convert the data file 240 so that it can be interpreted by the printer 20.

The type of conversion process to be performed by the printer driver engine 230 and other of its general
25 functions are under the control of I/O interface 200. This I/O interface 200 is a set of software instructions which attend to input and output to and from the user

(via the keyboard 16, mouse 18 and the display 14 previously described) and pass control parameters to the printer driver engine 230 in accordance with the user's input. The I/O interface performs the above operations
5 in accordance with two predefined groups of data, the first of which is a group of element definitions 220 and the second of which is a group of page definitions 215, both of which will be described more fully below. In overview, however, elements contain the control
10 information for the printer driver engine whereas pages contain information on how that control information should be displayed to a user.

Further details of the element definitions 220 will be
15 described with reference to Figure 3 which shows a number of such elements. As will be seen from the Figure these element definitions are interconnected in a hierarchical manner. A number of the elements (which can conveniently be termed branch elements) act as a passage to one or
20 more children, an example of such an element being colour/black ink element 40. Elements at the bottom of the hierarchy (which can conveniently be termed leaf

elements) determine a printer action, that is to say they are effective to control the printer driver engine 230.

25

In Figure 3, the home element 25 is positioned on the far left of the Figure and represents the overall "parent"

element. In the embodiment, the home element 25 has a single state and has four children, namely a colour/black ink element 40, a halftone element 60, a resolution element 65 and an ordering element 70. All these
5 elements have multiple states as indicated by a numeral in square brackets in the element.

Colour/black ink element 40 is effective, via its children, to set the ink type options for the printer 20,
10 that is to say whether the printer will print using plural colour inks or only black ink.

Both states of the colour/black ink element 40 are shown explicitly in the Figure from which it will be seen that
15 the children of this element are dependent upon its state. The element 40 has three children in its colour ink state (elements 41-43) and two (different) children (elements 47 & 48) in its black ink state. In the colour state, these children are brightness element 41,
20 contrast element 42 and saturation element 43. Of these three elements, the brightness element 41 and contrast element 42 are leaf elements in all states as they have

no subsequent children. Saturation element 43, has two states, namely manual (state 0) and automatic (state 1).
25 In its manual state it has three children, yellow element 44, magenta element 45 and cyan element 46. In its automatic state it has no children. It will thus be

noted that an element may be of a hybrid nature, that is to say, a branch element in some states and a leaf element in others.

- 5 It will be appreciated that certain elements have a large number of states (eg the colour contrast element which has 50 states). In the embodiment, this is displayed by presenting a graphic the content of which varies as it is clicked on, ie as the states are cycled through.
- 10 However, in a modification, such an element could be implemented by fewer states each representing intermediate values such as 25%, 50%, 75% and 100% (0% being in this case of no practical use) with a further state for a custom setting which may be entered via the
- 15 keyboard. Such a modification may be preferable in the case of an element having a high number of states in particular one exceeding 100 states to avoid a user needing to click on an element an excessive number of times. Furthermore, as a alternative to keyboard entry
- 20 of the custom setting, this could have a child element having the multiple states so that no recourse to the keyboard is necessary.
-

The Halftoning element 60 is used to allow the setting

25 of various halftoning options (e.g. ordered dither, error diffusion and the like). The Resolution element 65 is used to allow the setting of various printer resolutions,

usually expressed in dots per inch (dpi) for example, 300 dpi, 600 dpi etc. The Ordering element 70 allows the ordering of printer orientated supplies via the internet (e.g. paper, ink and the like).

5

This is achieved in the embodiment using identification information stored in the printer driver which is sent to the supplier when the ordering of supplies is selected. This identifies the type of printer as well
10 as relevant configuration information for that specific printer (eg whether it has been upgraded in any manner) Additionally, the printer driver may be pre-programmed with the address of a suitable supplier which may be the manufacturer.

15

As the device driver is peripheral specific, these supplies can be specific to the peripheral. This information, to give an example, will ensure that the correct ink for the printer will be ordered. Furthermore,
20 this facility will also allow a user to order new printer models from the manufacturer or to upgrade hardware or software. In the case of software upgrades this may be
downloaded immediately over the Internet.

25 The above elements have an arrangement of children dependent upon their state in a similar manner to the colour / black ink element and thus will not be described.

in detail. It will be appreciated that a number of other different printer options may also be accessible via appropriate elements, the four elements listed above being exemplary only.

5

Each of the elements shown in Figure 3, has its own element definition which contains details of its logical position relative to other elements that is to say it defines the above described hierarchical structure. The definition also contains the control information for the printer driver engine 230. This element definition will now be described with reference to Figure 4 which shows by way of example the definition for the colour/black ink element 40.

15

The element definition comprises two portions a basic definition portion 110 and a state definition portion 120, 130. In colour/black ink element 40 there are two states and therefore two state definitions, one for colour ink (state 1) 130 and another for black ink (state 0) 120.

In this embodiment, the basic definition portion 110 contains three pieces of information. The first is information concerning the parent of the element, which in the case of this element is the printer home element 25. The second piece of information is termed the "state

ribbon" and contains a pointer to a series of images for display to the user. As described more fully below, this allows an image indicating the present state of the element to be presented to the user. This series
5 contains an image for each state of the element in a sequential order. Thus in this case, there are two images, one for state 0 (black ink) and the other for state 1 (colour ink). The final piece of information is the default state of the element being that which the
10 printer driver is set upon first activation, in this case, the colour state.

As previously described, this element has two states and thus two state definition sections 120, 130. In the
15 black ink definition 120 (state 0), is information defining the "base page" associated with that state, in this case ink-options-black. This base page information is used in determining the user interface display in a manner which will be described later. Also defined in
20 the state definition 120 are the children associated with that state, in this case brightness element 47 and contrast element 48. As will be appreciated, each of

these children will have an element definition of its own.

25

Similarly in the colour ink state definition 130 comprises information defining its base page (ink-

options-colour) and its children (C-brightness element 41, C-contrast element 42 and saturation element 43).

It should be noted that the element definition does not
5 differentiate whether these children are leaf elements
or whether they have subsequent children themselves.

In a leaf element (e.g. brightness element 47) the state
definition will not include a list of children but
10 instead will include command data to be passed to the
printer driver engine 230 corresponding to the action to
be performed.

As has been explained above, the user interface structure
15 is defined by a group of element definitions. These
definitions are, however, separate from a group of page
definitions which define how the information will be
presented to the user on the VDU 14.

20 These page definitions will now be described with
reference to Figures 5 and 6.

Figure 5 shows the home page of the user interface, being
that which is displayed to a user upon selection of the
25 printer driver. The display comprises essentially of two
parts and is in certain respects similar to a web
browser. These two parts are the tool bar 36 for

navigating around the driver and the browsing space 34 in which a variety of elements can be displayed. The tool bar 36 has a backward arrow 36a and forward arrow 36b for navigating through the hierarchical structure of the driver and home control 36c which will return the browsing space 34 to the home page. In the Figure, back arrow 36a is presently deactivated because the browsing space is presently on the home page. The arrow is therefore greyed which is a known technique to indicate that the option cannot be selected.

Cursor 100 can be moved around the browsing space 34 by movement of the mouse 18. In the browsing space 34 are images representing the functional elements which can be accessed, in this case, colour/black ink element 40a, halftoning element 60, quality resolution element 65 and order supplies element 70. As will be seen, three of these are elements which have multiple states and an image corresponding to the current state is displayed. Two static elements, printer image 25 and logo image 27 which cause no function to be performed but are definable by the device manufacturer and may therefore be tailored to its house style are also displayed. In the figure the images are shown diagrammatically as areas having printed labels but they will preferably be images indicative of the function. For example, the image showing the state of the colour element 40a could display a rainbow of

colours or a pie chart showing different colours or the like.

Figure 6 shows the page definition of the home page shown
5 in Figure 5. This definition includes a list of elements
to be displayed, the list including the name of the
element, the X, Y coordinates at which it should be
displayed on the VDU and whether that element is static,
active or greyed. It will therefore be appreciated that
10 any number of elements can be displayed on each page and
each element can be placed at any position on the screen.
As previously explained, each element may be specified
as static, active or greyed for that particular page.
A static or greyed element is one which is merely
15 displayed but causes no action to occur. An active
element is one which may be selected by the user and may
act as a gateway to its children or cause an action to
take place. It should be appreciated, however, that
greyed and static elements are not equivalent. A static
20 element remains so on all pages whereas a greyed element
is only non-selectable on particular pages and will be
active on other pages. Greyed elements are displayed by

fading their image, that is to say reducing the image
saturation of the appropriate area of screen.

25

In Figure 6 the first listed element is the company logo
element 27 which is to be positioned at position X = 5,

Y = 5 on the page, it being noted that, in the embodiment, the X coordinate is measured from the bottom left of the page and Y coordinate from the top left. The company logo element 27 is a static element. The next
5 element is the printer home element 25, is to be printed at position X = 10, Y = 85 and is also to be a static element. The third element on the page is the colour/black ink element and is to be displayed at position X = 55, Y = 10 and is to be an active element,
10 that is to say that it may be selected by a user and will cause an event to occur as will be described below. Likewise there are active elements positioned at defined places for the halftone element 60, the resolution options element 65 and the order supplies element 70.

15

The working memory 210 of the I/O interface of the printer device 199 will now be described with reference to Figure 7. An identification number of the present page, i.e. that currently displayed on the screen, is
20 stored in a first memory location 211. In a second location 212, an identification number of the presently selected element is stored. Finally, in a third group of memory locations 213 one for each element, the current numeric state of each element is stored.

25

The operation of the I/O interface 200 of the printer driver 199 will now be described with reference to the

flow diagram of Figure 8. In step 300, the home page is read and displayed to the user on the VDU. In step 302, the I/O interface 200 detects whether the user has clicked on an image thus selecting an element. In the
5 embodiment, the cursor 100 will change its appearance to the user so as to indicate that it lies above an element which may be activated (i.e. not above a static or greyed element or an unused portion of the screen).

10 Once the user has selected an element, in step 304 the I/O interface determines whether the type of selection is one indicating that the user wishes to change the state of the selected element or instead whether the user requires expansion of the element to display the elements
15 available further down the hierarchical structure. As will be described more fully below, the I/O interface 200 distinguishes between the above two possibilities in dependence upon the type of mouse button click (single left click, double left click or single right click). It
20 also takes into account the relationship between the selected element and the presently displayed page (that is to say whether the selected element is a parent or

child type element in relation to the other elements on that page).

25

In the case where the user has selected an element for a change of state, in step 306 the I/O interface 200

changes the state of that element and updates the working memory 213. The new state may be a single increment from the previous state (cycling round to the first state again if the previous state was the numerically highest state) or may be selected precisely by the user depending upon which of the mouse buttons were used to make the selection as will be described more fully below. In step 308, the I/O interface 200 determines how the display should be changed. This may require a new page to be displayed or merely the updating of the current page, fuller details of which are described below. In step 310, if there is a printer control command associated with the new state selected then this is output to the printer driver engine 230. The new or updated page is then displayed in step 314 and the page pointer 211 updated if necessary. The user interface then returns to step 302 to await the selection of the next element.

In the alternative situation where an element has been selected for expansion the process branches to step 312 where I/O interface determines the new page to be displayed which will be that which corresponds to the expanded element. This new page is determined by reading the basepage details corresponding to the present state of the element from its element definition as will be described in greater detail below. The new page is then displayed at step 314 and the page pointer 211 is

updated. The process then returns to step 302 as before.

In order to explain the above described process more fully, the process steps performed by the I/O interface upon three different categories of user selection of an element will now be described. These three categories are a single left mouse button click, a double left mouse button click and a single right mouse button click.

10 Figure 9 shows the process steps carried out by the I/O interface upon the single click of the left mouse button on a selected element.

At step 400 the single click of the left mouse button is
15 detected. At step 402, the I/O interface detects whether the cursor 100 lies over a static element. If it does lie over a static element (or a greyed element), the process branches to step 414 as the process is considered complete. If an active element has been selected, the
20 process branches to step 404 where the I/O interface determines whether the selected element in its current state is a leaf element. This is achieved by reading the

portion of the element definition 220 of the selected element corresponding to its presently selected state.
25 If this state definition has no children then the element is a leaf element.

Describing first the situation where the element is not a leaf element, the flow branches to step 406. In step 406, the base page is read from the current state of the selected element. The page listed as the base page will
5 contain the children of the element selected for expansion. Once the new page to be displayed has been read, in step 408 this is displayed on the VDU and the page pointer 211 is updated.

10 An example of the above operation, from the point of view of a user, will now be described. In this example, it is first assumed that the presently displayed page is the home page shown in Figure 5. It is further assumed that the user has located cursor 100 to lie on top of colour
15 ink element 40a and has selected this using a single click of the left mouse button (step 400). Following the process described above, the I/O interface will read at step 404 the element definition for the ink options element (shown in Figure 4) and thus determine that the
20 presently selected state has children. The process will thus proceed to step 406 where the base page of the present state of the selected element will be read which,
in this case is named ink-options-colour. Thereafter the contents of the ink options colour page (not shown) will
25 be read and page will be displayed. In this case, a page corresponding to Figure 10 will be displayed.

Explaining Figure 10 in greater detail, tool bar 36 and browsing space 34 are as previously described. The four elements 40a, 60, 65, 70 on the right hand side of the browsing space 34 in Figure 5 are now presented at the left of the browsing space 34. As an optional feature, a scrolling movement may be used to effect this. Children of the element 40a and, in this embodiment, grandchildren have now been displayed on the screen. Children brightness element 41 and contrast element 42 which are both leaf elements are displayed. Also displayed is saturation element 43 and (as manual saturation control is currently selected) grandchildren elements yellow 44, magenta 45 and cyan 46 are additionally displayed. As the selection of colour is currently of interest to the user, the elements 60, 65 and 70 which remain on the browsing area in this example are greyed, meaning that they cannot be directly selected from the screen. As will be appreciated, a return to the home screen of Figure 5 will be necessary in order to select these elements which can be achieved using the back arrows 36a or the home icon 36c. It should be noted that the parent page in the element definition allows the back arrow feature to be implemented.

Returning to Figure 9, the process where the selected element is a leaf element will now be described. In such a event, the process will branch to step 410. In this

case the user will have selected the element in order to change its state. The user interface, therefore, reads the current state of the selected element from the state definition block 213. This is then incremented by one
5 to the next numerically higher state. In the event that the numerical value of the state exceeds the number of possible states, it is reset to state zero. Thus the state selection is cyclic in nature. As will be apparent, in the case of an element having only two
10 states, this effectively achieves a toggling between the two states. The numerical value of the new state is then stored in the appropriate memory location 213. In step 412, the appearance of the new page is determined. The page to be displayed is already known and stored in
15 location 211. As previously described, that page definition includes the position of all the elements. However, as a state change has occurred, an updated image must be displayed corresponding to the selected element. This is read from the state ribbon of the selected
20 element. As previously described, this state ribbon contains a series of sequential images, one for each
~~numeric value of the state. In this manner, the page~~
definition defining the elements to be displayed will be updated to display the image of the element in the
25 presently selected state. The process also determines, from the current state portion of the selected element whether an action is to be performed and, if so, performs

that action. The process ends at step 414.

The process steps when an element has been double clicked using the left mouse button will now be described with reference to Figure 11. This is commenced at step 500 by the left mouse button double click. At step 502 it is detected whether the element is static (or greyed) and if so the flow branches to step 520 where the process ends. Otherwise, at step 504 it is detected whether the element is a leaf element or not in the manner previously described. If the element is a leaf element, then its state is changed at step 506, and updating performed in the same manner as that employed for a leaf element when a single click has occurred which will not, therefore be explained again.

In the event that the selected element is not a leaf element, the flow branches to step 510 where it is detected whether the current page is the base page for the element. This is necessary because an element may be displayed either with its parent or its children. In the event that it appears with its children it will be necessary to change the page with the change of state so as to display the new children. On the other hand, if the selected element appears with its parent it is only necessary to update the page with the new state of the selected element. The processes by which these two

options are achieved will be described in greater detail below.

In order to determine whether the current page is the
5 base page for the selected element, the user interface
reads the basepage from the element definition of current
state of the selected element and compares this with the
present page stored at memory location 211. In the event
that the current page is not the base page, the process
10 branches to step 516 and increments the state of the
element. As previously explained, it is cycled to state
zero if necessary. In step 518, the current page is
updated with the new state image for the element using
the state ribbon for the new state as previously
15 described.

Again using the home page shown in Figure 5 as an example
this process will be described as it appears to a user
in this case, a double click on colour ink element 40a
20 is an indication that the user wishes to change the state
of this element rather than expanding it. The process
~~will have branched to step 510 where it will have~~
determined that the present page does not correspond to
the base page of the selected element (see reference
25 numeral 130 of Figure 4). The process will thus have
branched to step 516 where the state of the element is
changed. In this case there are only two states and so

this state is cycled to state zero. The image for the element is read from the state ribbon and the updated page is then displayed on the browsing space 34 of the VDU. Figure 12 shows the displayed output on the VDU
5 following the above process.

As shown in Figure 12, the displayed output is similar to that of Figure 5 except that a black ink element image 40b is displayed instead of the colour image element 40a.
10 As will be appreciated, the expansion of black ink element 40B will subsequently give different children options to those which would have been presented had the alternative state of colour image of 40a been selected.

15 Returning to Figure 11, if at step 510 it is determined that the page is the base page for the element then the process branches to step 512 where the state of the element is incremented in the same manner as described above. In this case, however, the new base page for the
20 new page is read from the state information of the selected element and it is this new base page which is displayed. The page pointer 211 is updated accordingly.

The process thereafter terminates at step 520.

25 Using the page displayed in Figure 10 as an example the above process as it would appear to a user will now be described. It is assumed that the user has double left

clicked on colour image element 40a. It will be appreciated that the current page will be the base page for that selected element. The process of Figure 11 will thus have branched to step 512 and changed the state of
5 the ink options element to black ink (state 0). The new base page will have been read and displayed. This will result in a user display as shown in Figure 13.

Figure 13 shows tool bar 36 and browsing space 34 as
10 previously described. To the left of browsing space 34 now appears black ink element 40b (in place of colour ink element 40a) below which are greyed halftoning element 60, quality/resolution element 65 and order supplies element 70 as before. On the right of the browsing space
15 34 are the two children of black ink element 40B namely monochrome brightness 47 and monochrome contrast 48 which replace the children of the colour ink element 40a.

The actions performed by a click of the right mouse
20 button will now be described with reference to Figures 14A and 14B. In overview, however, a similar process is followed to that of a single left mouse click except that

instead of the present state of the element being incremented by a single state, the user may select the
25 new state of the element from a list of all possible states using the mouse. In the embodiment, the current state of the element is indicated by a tick adjacent to

it.

Referring to Figure 14A at step 600 a single right click of the mouse button is determined. At step 602, it is
5 determined whether the element selected is a static element in which case the process ends at step 636. Otherwise, at step 604 it is determined whether the selected element is a leaf element. The manner of this detection has been previously described and will not be
10 repeated.

If the element is a leaf element, the process branches to step 606 where the context menu for that element is displayed. A context menu is a portion of the displayed
15 screen which temporarily over writes the previous image and contains a list of all possible states for that element. In a modification, however, the space allocated to the context list may be of predetermined size and may be scrollable if the number of state images are too great
20 although this is considered non-preferable. Continuing to Figure 14B, the context menu is highlighted so as to indicate the current state. In the embodiment this is

by means of tick adjacent to the image but other forms of such indications could be used (e.g. by making the
25 image flash or larger than the other state images). At step 610 it is detected whether the user wishes to change the state of the element. The user does this change by

moving the cursor to the desired new state and confirming the selection by clicking a mouse button, this being a known procedure for choosing an item from a menu. If the user does not change the state of a element then the process ends at step 616. If the user does change the state of the element then updating steps (steps 612 and 614) as previously described with reference to a left mouse button click are performed. At step 616 the process terminates.

10

Returning to step 604, if the element is not a leaf element then a context menu for the selected element is displayed at step 618 in a similar manner to the context menu of a leaf element. Again current state of the element is indicated in step 620 using a tick adjacent to the current state of the element. At step 622 it is determined whether the user wishes to change the state of the element and if no change is required, the process branches to step 634 and the process ends. Otherwise, the process branches to step 624 where it is determined whether the current page is the base page for the element in the same manner as that previously described (see step

510, Figure 11). If the current page is not the base page for the element then the state of the element is changed to that selected by the user (step 630) and the current page updated with the new state image for the element is displayed at step 632 using the state ribbon

information as previously described. The process thereafter ends at step 634. Conversely if at step 624 it is determined that the current page is the base page for the element then the state of the element is changed
5 to that specified by the user at step 626 and, at step 628, the base page of the new state is displayed. This is done in the same manner as described above. The process thereafter ends at step 634.

- 10 The other options presented in the tree diagram of Figure 3 will not be discussed in detail as the architecture and functionality is the same as that described above. However, with reference to element 70, it will be appreciated that the printer device driver can be used
15 in conjunction with a modem in order to place orders directly with a supplier for consumables such as paper, ink cartridges or the like. Furthermore, the architecture of the present invention allows the images of these elements to look like the consumables on offer.
20 Thus the user may be presented with a picture of a colour ink jet cartridge and, merely by clicking on that cartridge, will be able to forward the selected article.
-

Indeed, the architecture of the present invention will allow an original equipment manufacturer to personalise
25 the device driver so as to maintain a house style with its products and its website.

In the embodiment, a driver for a colour ink jet printer was described. The invention may, however, be used to drive any type of printer such as ink jet printers in general, laser printers, dye sublimation printers or
5 thermal wax transfer printers.

In the embodiment, a printer driver was described. The invention is, however, suitable for all types of device drivers and could, therefore, be employed in device
10 drivers for modems, scanners, monitors or the like.

In the embodiment described above, a state ribbon has been used in order to define the images corresponding to each state of an element. This need not be the case and
15 the state definition portion of each element could contain the image information. An advantage with the state ribbon is simplicity but there is a disadvantage that all of the images must be of the same size. The use of individually defined images for each state allows
20 different size images to be used for different states.

In the embodiment, a single image has been displayed to represent and indicate the state of an element. However the invention is not so limited and the displayed image
25 may, in fact, be animated and thus comprise a series of images so as to present a moving image to the user. Additionally sound may accompany the image. Alternatively

the image representing the state of an element could be handled by a "plug in" which is an external piece of software which functions or appears to function as an integral part of the system and takes that responsibility
5 for implementing a plurality of additional features for an on behalf of the system, those features appearing to the user as if they were provided by the system itself.

In the embodiment specific events have been triggered by
10 specific mouse button clicks. The described relationship is not intended to be limiting and other mouse events could be tied to the triggered event. Furthermore a three button or wheel-mouse could be used as could any other type of pointing device.

15

In the embodiment, the size of each image to be displayed is dependent upon the stored bit image. However, it would be possible to include a scaling parameter into the element definition so that the same bit image could be
20 used for different pages but displayed in different sizes. For example, when the element is a parent the size could be larger than when it is child. Scaling of

images may also be necessary in order to improve the presentation of the elements on a page.

25

A further modification is the introduction of a cursor which can be customised. As previously explained, it is

known to change the appearance of the cursor from a first appearance when it does not lie over an activatable element to a second appearance when it does lie over an activatable element. A new image for the cursor when it
5 lies over an activatable element could be stored in the element definition. This would allow a different cursor to appear dependent upon what type of element the cursor was lying over. An example of a suitable element would be an ink pot when the cursor lay over the ink options
10 element.

A further alternative is the provision of a tool known as an agent within the printer driver. Agents are an
15 enhanced interface between a user and system help information but are not currently in use in device drivers. Agents may take the form of an animated character to whom users may ask questions in a natural language in order to obtain technical information and
20 help on the system. In the case where the invention is embodied in a printer driver, a suitable agent would be a character constructed from a simplified stylised image
of a printer.

25 In a further modification to the embodiment, the printer driver would monitor the remaining amount of consumables, such as the amount of ink or paper. Upon this falling

below a preset amount, the printer driver would cause a prompt to be displayed to the user requesting the user to indicate whether a replacement consumable should be ordered. In a further alternative, the driver could
5 order a replacement consumable automatically, without the need for user confirmation.

In a further modification, the driver would keep a record of the age of the driven printer and would call the
10 attention of the user to new product after a pre-determined passage of time (eg two years). Information on these new products would be downloaded from the internet.

15 In the embodiment, each display screen consists of parents on the left of the browsing space and children on the right allowing a left to right scrolling effect to be performed upon navigation through the elements. However, this need not be the case and the image
20 representing the home element or the present parent could be in the centre of the browsing space with the children arranged in a circular fashion around it. In fact, any

type of page display may be achieved merely by customisation of the page data.

25

In the above description, elements having one state have not been described in detail. One state elements would

be leaf elements which perform a single function, e.g. nozzle cleaning. The selection of an element would cause the I/O interface 200 to instruct the printer driver 230 directly to perform the function.

5

Other modifications and variations will be apparent to those skilled in the art.

CLAIMS

1. A device driver for interfacing a computer to a peripheral, the device driver comprising:

5 a device driver engine (230) for driving the peripheral in accordance with control commands;

means (220) for defining a plurality of elements, each element having at least one state, each state having an associated image for display to a user for user
10 selection and at least one of a pointer to one or more associated elements or a control command to be output to the device driver engine;

means (215) for defining a plurality of pages, each page being associated with a state of an element and
15 listing at least one of said one or more associated elements; and

means (200) for (i) receiving a user selection of a current state of a current element; (ii) reading the page associated with the current state of the current
20 element; (iii) generating a display using the associated image of each element identified in the page list for the current state of the current element; and (iv) outputting
the control command, if any, of the current state of the current element to the device driver engine.

25

2. A device driver according to claim 1 wherein said means (200) for generating a display using the associated

image of each element identified in the page list for the current state of the current element generates a display using the associated image for the current state of the each identified element.

5

3. A device driver according to any preceding claim adapted for the driving of a printer.

4. A method of interfacing a computer to a peripheral,
10 the method comprising the steps of:

providing an engine for driving the peripheral in accordance with control commands;

defining a plurality of elements, each element having at least one state, each state having an
15 associated image for display to a user for user selection and at least one of a pointer to one or more associated elements or a control command to be output to the device driver engine;

defining a plurality of pages, each page being
20 associated with a state of an element and listing at least one of said one or more associated elements; and
receiving a user selection of a current state of a

current element;

reading the page associated with the current state
25 of the current element;

generating a display using the associated image of each element identified in the page list for the current

state of the current element;

outputting the control command, if any, of the current state of the current element to the device driver engine.

5

5. A method according to claim 4 wherein said step of generating a display using the associated image of each element identified in the page list for the current state of the current element generates a display using the associated image for the current state of the each identified element.

10

6. A method according to any preceding claim adapted for the driving of a printer.

15

7. A printer driver for interfacing a computer to a specified printer, the printer driver comprising:

means for inputting a user command to order supplies associated with said specified printer; and

20 means for causing the computer to output a signal addressed to a supplier indicative of the user's desired supplies.

8. A printer driver according to claim 7, wherein said means for causing the computer to output a signal is adapted to output said signal to the internet.

25

9. A printer driver according to claim 7 or 8 further comprising means for presenting the user with the options to order at least one of:

an ink cartridge specific to the specified printer
5 or a process cartridge specific to the specified printer.

10. A method of generating signals for ordering printer specific supplies comprising the steps of:

providing a printer driver of the type used for
10 interfacing a computer to a specified printer;

inputting a user command to order supplies associated with said specified printer; and

causing the computer to output a signal addressed to a supplier indicative of the user's desired supplies.
15

11. A method according to claim 10 further comprising the step of presenting the user with the options of ordering at least one of an ink cartridge specific to the specified printer or a process cartridge specific to the specified printer prior to the input of the user command.
20

12. A method of inputting commands to a device driver, said device driver interfacing a computer with a peripheral, the method comprising the steps of:

25 providing a plurality of element definitions, said elements being interconnected in a hierarchical manner, each element having at least one state, each state

defining a display image and at least one of a pointer to one or more associated elements or a control command to be passed to the device driver;

providing a plurality of page definitions, each of
5 said pages being associated with a state of an element and each said page definition containing a list of elements;

reading the list of elements from the page definition of the presently selected page;

10 displaying the display image corresponding to the present state of each of said elements on the list on a display device;

inputting a user selection of a displayed element;

upon a first type of user selection, reading the
15 page associated with the present state of the selected element and displaying the display image for the present state of each of said elements on the list of that page definition;

upon a second type of user selection (i) changing
20 the state of the selected element, (ii) determining whether there is a page definition associated with the new state of the selected element and if so reading said page definition and displaying the display image for the present state of each of said elements on the list of
25 that page definition; or if not displaying the present page with the display image of the selected element updated so as to correspond to the new state.

13. A printer driver for interfacing a computer to a printer, the driver comprising:

means for driving an internet browser style user interface so as to cause the display of user definable printer parameters and so as to receive user definitions;

means for storing information specific to the printer to be driven;

means for receiving a purchase command from the user via said internet browser style user interface; and

means for outputting purchasing data to the Internet for ordering the supply of a printer related item from an external supplier in dependence upon said received purchase command and said stored information.

14. A printer driver for interfacing a computer to a printer, the driver comprising:

means for displaying user definable printer parameters and for inputting commands from a user;

means for storing information specific to the printer to be driven;

means for receiving a purchase command from the user via said input means; and

means for outputting purchasing data for ordering the supply of a printer related item from an external supplier in dependence upon said received purchase command and said stored information.

15. A printer driver according to claim 13 or 14, wherein said purchasing command relates to a printer consumable.

5 16. A printer driver according to claim 15, wherein the driver further comprises means for monitoring the current amount of the printer consumable and means for prompting the user to consider a purchase command upon the current amount of the printer consumable falling below a
10 predetermined value.

17. A method of ordering printer specific consumables comprising the steps of:

providing a printer driver of the type used for
15 interfacing a computer to a specified printer;

monitoring the current amount of at least one printer consumable; and

causing the computer to output a signal addressed to a supplier so as to order a replacement consumable
20 upon the current amount of the consumable falling below a predetermined value.

18. A printer driver for interfacing a computer to a
25 printer, comprising:

means for monitoring the current amount of at least one printer consumable; and

means for causing the computer to output a signal addressed to a supplier so as to order a replacement consumable upon the current amount of the consumable falling below a predetermined value.

5

19. A storage medium carrying computer readable commands for causing a computer to perform a method according to any one of claims 4 to 6, 10, 11, 12 or 17.

10

20. A signal carrying computer readable command for causing a computer to perform a method according to any of claims 4 to 6, 10, 11, 12 or 17.

21. A printer driver having a web-style user interface.

Abstract:Graphical User Interface

5 A device driver is provided in which the user
interface is arranged so as to appear in a similar
fashion to that of an internet browser. From within the
device driver, products relating to the driven device may
be ordered preferably via a message sent over the
10 internet. The internal structure of the driver is
arranged as two groups of data defining pages and
elements allowing versatility in the design of the
driver. The invention has particular application in the
field of printer drivers.

15

20 Fig. 2

ACT/GB90/03/19
20/00
Beresford & Co

This Page Blank (uspto)
